Pulsed-Latch Circuits: A New Dimension in ASIC Design

Youngsoo Shin
KAIST

Seungwhun Paik
Synopsys

Pulsed-latch circuits retain the advantages of both latches and flip-flops, offering higher performance and lower power consumption within a conventional ASIC design environment. This article identifies a design methodology and tools for pulsed-latch ASICs to complement this environment. The authors review potential solutions and provide quantitative results to assess the effectiveness of pulsed-latch circuits.

■ **A conventional ASIC** design mainly uses an edge-triggered flip-flop as a sequencing element because of the simplicity of this flip-flop's timing model. Specifically, the amount of time available to a combinational block that lies between two flip-flops is fixed. This constrains timing uncertainties within each combinational block, which is important for design steps at higher abstraction levels such as logic synthesis when implementation details are unknown. However, an appreciable portion of the clock period, total power consumption, and circuit area are attributed to flip-flops. A typical flip-flop has 6 FO4 (fanout-of-4) delays (sum of the clock-to-Q delay and the setup time).[1] This is 13% of a 46-FO4-delay clock period, and 17% and 21% when the clock period becomes 35 and 29 FO4 delays, respectively,[2] due to the requirement to increase the clock frequency. The clock distribution, including flip-flops, often contributes more than half of the total power consumption.

High-performance custom designs often use a level-sensitive latch as a sequencing element. Its timing overhead ranges from 2 to 4 FO4 delays,[1] which is far smaller than that of a flip-flop. Such designs are somewhat immune to clock skew and jitter, owing to latch transparency. However, the timing model is more complicated, which, along with the limited support of CAD tools, makes using latches in ASIC designs difficult. In addition, data must be held for a longer period of time, increasing the likely number of hold time violations.

A *pulsed latch* is a latch that is driven by a brief clock pulse. The amount of time available to a combinational block is still variable, but the amount of variation is significantly less than in latch circuits. The scope for hold time violations is also reduced. This makes a pulsed latch an ideal sequencing element for high-performance and low-power ASIC designs, as well as for traditional high-performance microprocessor designs.[3-9]

Given the similarity of the timing model, a pulsed latch can be approximated by a faster flip-flop. This enables a simple migration of a flip-flop circuit to a pulsed-latch version by substituting all (or some) of the sequencing elements,[10] thus saving both the clock period and power consumption. We have performed experiments using some test circuits to examine the savings quantitatively. Additional support of the design methodology and tools is necessary for a complete environment of ASIC design based on pulsed latches.

In this article, we discuss this additional support for

- the physical design, including insertion of pulse generators and customized placement algorithms;
- timing analysis and optimization, including buffer insertion to resolve hold time violations, time borrowing via different pulse widths, and mixed time borrowing and sequential optimization; and
- low-power design, including clock gating in pulsed-latch circuits.

0740-7475/11/$26.00 © 2011 IEEE Copublished by the IEEE CS and the IEEE CASS

Pulsed-latch circuits

In a pulsed-latch circuit, a normal clock is delivered from a clock source to multiple pulse generators (called *pulsers*), which are dispersed in a placement region. Each pulser then delivers a pulse to more than one nearby latch, because a pulse can be easily distorted over a long distance.

Researchers have proposed several implementations of pulsers.[4,8,11] For example, when the pulser shown in Figure 1a is implemented in 45-nm technology, it consumes about 5 times more power than a standard latch because of the large number of clocked transistors, even though it takes only 30% more area. Reducing the number of pulsers, therefore, is important for the power consumption.

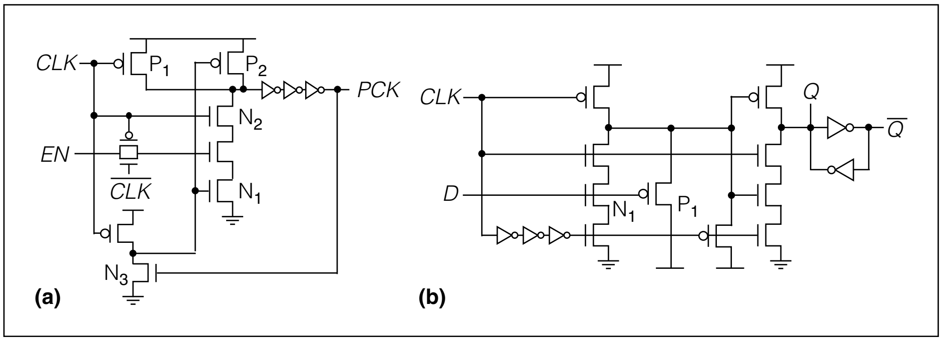

Figure 1. Example of a pulser[4] (a), and a pulsed flip-flop[3] (b). In (a), when clock *CLK* is 0, PMOS transistor P_1 (as well as NMOS transistor N_1) turns on and causes pulse clock *PCK* to be 0. The pull-down network is enabled after N_2 is turned on at the rising edge of *CLK*, and *PCK* is driven high. N_3 and P_2 are subsequently turned on, which drives PCK back to 0. Thus, the pulse width is determined by the inverter chain delay. Notice that clock gating, driven by signal *EN* (enable) is embedded in the circuit; *EN* is not allowed to change while *PCK* is 1, which occurs briefly, and can be realized by a simple transmission gate.

A pulser can be embedded in a latch, known as a *pulsed flip-flop*, to avoid distortion of the pulse shape. Figure 1b shows an example of such a circuit element; a pulse is not explicitly generated, but the inverter chain delay determines the period of time that data can be captured; this period of time is thus equivalent to the pulse width. The integration of a pulser and a latch comes at the obvious cost of more area and power consumption than in a shared external pulser, but the benefits of a pulsed latch include less sequencing overhead, a simple timing model, and only a small amount of time borrowing. The circuit in Figure 1b also allows for integrating a logic gate by modifying NMOS transistor N_1 and PMOS transistor P_1, as is often done in high-performance processor designs.[5,12]

To take advantage of both pulsed-latch (shared pulser) and pulsed flip-flop (no distortion of pulse shape), we can integrate more than one latch with a single pulser. This yields a *pulsed register*,[4] which can be used in data path circuits.

Design of pulsed-latch ASICs

In its simplest form, a pulsed latch is like a faster flip-flop with a longer hold time. We can migrate a conventional ASIC design synthesized with flip-flops to a pulsed-latch version by simply replacing all the flip-flops with latches. We must insert and properly place a necessary number of pulsers to guarantee the integrity of the pulse shape. We should use timing analysis to check for a likely increase in hold time violations and remove these violations by inserting delay buffers. This simple migration allowed roughly a 10% decrease in the clock period and a 7% decrease in power consumption in our experiments using ISCAS-89 benchmark circuits and open-core circuits (http://www.opencores.org) with 45-nm technology, demonstrating the benefit of pulsed-latch circuits.

A pulsed latch can be used progressively if its pulse width is modulated, implying the existence of pulsers generating more than one pulse width. The resulting difference between the pulse widths of the launching and the capturing latches can be used for time borrowing. The problem of assigning a pulse width to each latch is called *pulse width allocation* (PWA). The amount of time borrowing from PWA is necessarily limited because increasing the pulse width raises the risk of hold time violations. This limitation can be alleviated, however, if PWA is used along with sequential optimization techniques such as clock skew scheduling and retiming; such an approach can decrease the clock period by an additional 20%.

When a pulsed-latch circuit is employed for low-power applications, applying clock gating is beneficial. If clock gating is implemented in each pulser, as in Figure 1a, a pulser must be connected to latches that are gated by the same enable signal, whereas the connection should be made in consideration of pulse shape. In other words, pulser insertion should

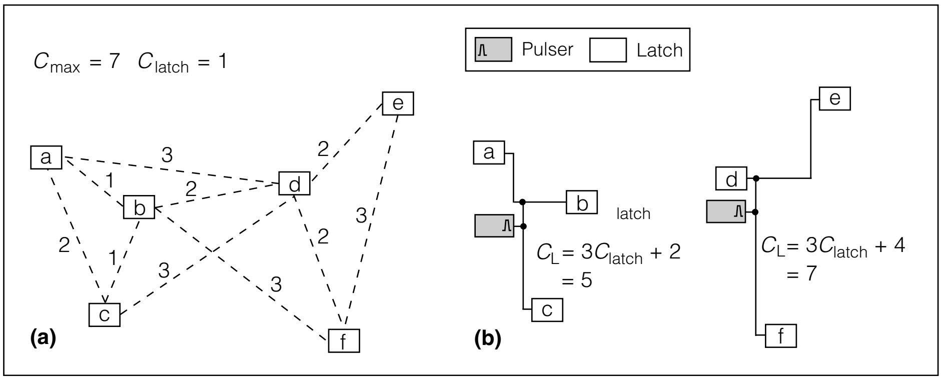
$C_{\max} = 7$ $C_{\text{latch}} = 1$
Pulser
Latch
a
b
c
d
e
f
3
1
2
2
2
1
3
3
2
3
(a)
(b)
latch
$C_L = 3C_{\text{latch}} + 2$
$= 5$
$C_L = 3C_{\text{latch}} + 4$
$= 7$

Figure 2. Location of latches with wire capacitance (a), and an example solution (b).

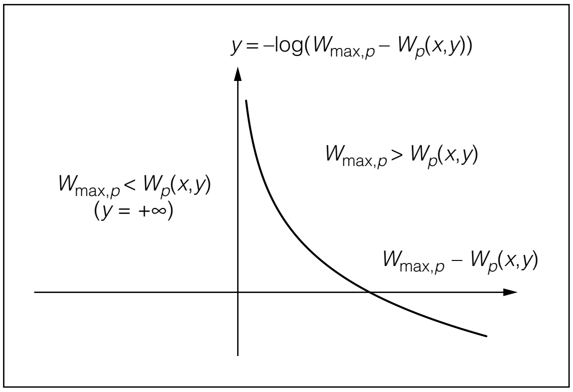

Figure 3. Definition of a log barrier in a global placement problem.

consider the functionality of latches as well as their physical proximity. An additional 15% decrease in power consumption can be expected after clock gating is applied.

Pulser insertion

Pulser insertion should be carefully performed because of the requirement of physical proximity between latches and pulsers. An arbitrary grouping of latches and inserting pulsers would likely yield a bad physical design, because latches in the same group would end up in a localized region, severely constraining the overall placement.

Pulser insertion could be done after the initial placement and latch locations are given; the problem is to find a minimum number of pulsers, such that each group of latches and their pulser (called a *pulser group*) satisfy the pulser's load capacitance limit ($C_{\max}$). This is illustrated using an example in Figure 2a, where C_{latch} indicates a latch's input capacitance. The connection between each pair of latches is associated with the wire capacitance; the connections that cause $C_{\max}$ to be violated are dropped. For example, the wire capacitance between a and e is larger than 5, which, when added to $2C_{\text{latch}}$, exceeds $C_{\max}$. Figure 2b shows an example solution.

A similar instance of the problem is to identify a group of latches that can be mapped to a single pulsed register, rather than latches and an external pulser.

Placement of pulsed-latch circuits

Once pulser groups are identified, the entire design should be placed again, either incrementally or as a completely new placement step. A heuristic method using a conventional placement tool is to assign a higher net weight to the nets connecting a latch and a pulser,[13] or to create a relative placement bound that encompasses each pulser group.

A systematic method is to explicitly consider the connection between a pulser and a latch during placement. In this direction, consider the following definition of a global placement problem:[14]

$$\begin{aligned} &\text{Minimize } W(x, y) \\ &\text{Subject to } D_b(x, y) \le D_{\max,b}, \quad \forall \text{ bin } b \\ &\qquad\qquad W_p(x, y) \le W_{\max,p}, \quad \forall \text{ pulser group } p \end{aligned}$$

where W denotes the total wire length, D_b is the density of bin b when the placement region is divided into a grid of bins, and W_p is the total wire length of pulser group p. $W_{\max,p}$ is the maximum wire length allowed in p, and is derived from the load capacitance limit of a single pulser and the total input capacitance of the latches in the group.

This constrained problem can be transformed to an unconstrained one:

$$\begin{aligned} \text{Minimize } & W(x,y) + \alpha \sum_b [D_{\max,b} - D_b(x,y)]^2 \\ & + \beta \sum_p -\log[W_{\max,p} - W_p(x,y)] \end{aligned}$$

where α and β are some constants. The use of log is important (see Figure 3 for its definition). When $W_{\max,p} - W_p(x, y)$ becomes negative, meaning the pulser group's total wire length exceeds its allowance, we define $+\infty$ for the value of $-\log[W_{\max,p} - W_p(x, y)]$

so that this placement can be rejected. The same happens as $W_p(x, y)$ approaches $W_{\max,p}$ on the right of the *y*-axis. *Legalization* (removal of any remaining overlaps of cells) follows the global placement.[13]

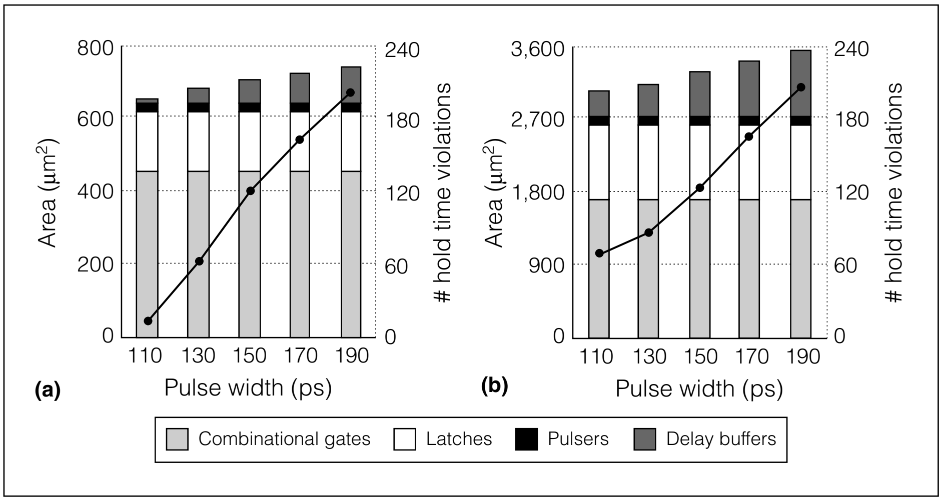

Figure 4. Number of hold time violations and extra buffers with different pulse widths for the s1423 ISCAS-89 benchmark circuit (a) and the usbc open-core circuit (b).

Timing optimization

Timing of pulsed-latch circuits can be further optimized if we modulate pulse widths (i.e., PWA). Combining PWA with a conventional sequential-optimization technique is even more effective in reducing the clock period. Hold time violations and variations in pulse widths also must be considered.

Resolving hold time violations

Pulsed-latch circuits have an increased risk of hold time violations. Data launched at the rising edge of a pulse must arrive at a capturing latch after the hold time is past the falling edge. Therefore, more hold time violations are likely to occur with increasing pulse width. We can correct these violations by inserting delay buffers[15-17] or using resynthesis[18] to increase the delay of the short paths.

Figure 4 shows the number of hold time violations, along with the area of the delay buffers (indicated by the hatched pattern) needed to correct them. We tried several pulse widths, with the most narrow being 110 ps. Clearly, the number of violations and corresponding number of buffers increased as the pulse became wider. Even at 110 ps, the proportion of the buffer area can be substantial in some circuits such as the usbc open-core circuit (http://www.opencores.org), signifying the importance of hold time violations in pulsed-latch circuits.

A proactive approach to avoid hold time violations is to explicitly consider hold time constraints during logic synthesis. In our experiments with test circuits, the difference between this approach and buffer insertion in total area turned out to be small, even though the netlists were very different.

Time borrowing

A small amount of time borrowing is possible in pulsed-latch circuits, even though a pulse is very short. This possibility would be deliberately ignored in ASIC design to simplify the timing model. Specifically, if the rising edge is regarded as the time when data is launched, the same edge is assumed for the time when data is captured.

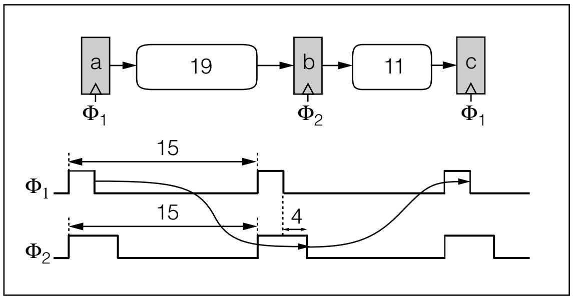

Figure 5. Time borrowing via multiple pulse widths. The delay between latches *a* and *b* is 19, and that between *b* and *c* is 11. The pulse applied to *b* (ϕ_2) is wider by 4 than that applied to *a* and *c* (ϕ_1). The period of both pulse clocks is set to 15.

If we use more than one pulse width, another form of time borrowing emerges, as Figure 5 shows. In this setting, the block between *a* and *b* effectively borrows 4 time units from the block between *b* and *c*, thereby working correctly even though its delay is larger than the clock period. Note that the clock period must be set to 19 if ϕ_1 is applied to all three latches.

This approach can be generalized to a problem in which the pulse width is allocated to latches (PWA) such that the clock period is minimized.[13] A list of pulse widths is defined by pulsers that are available;

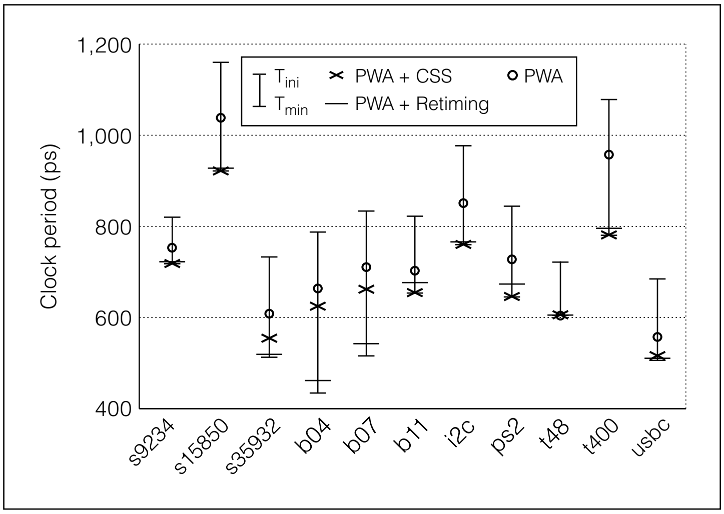

Figure 6. Comparison of pulse width allocation (PWA), mixed PWA and clock skew scheduling (CSS), and mixed PWA and retiming.

the maximum pulse width is necessarily limited because of the increasing number of hold time violations. The PWA problem can be readily solved through an iterative relaxation method.[19]

Mixed time borrowing and sequential optimization

Figure 6 shows the result of applying PWA to some circuits. The figure also shows the clock period of the initial pulsed-latch circuit (T_{ini}) and the minimum clock period (T_{min}), which is obtained through standard clock skew scheduling (CSS).[20] It is clear that PWA alone cannot achieve the minimum clock period when three pulse widths (130 ps, 190 ps, and 250 ps) are assumed.

We can alleviate this problem by combining PWA and clock skew scheduling (CSS),[13] so that the two techniques complement each other. In CSS, realizing a large amount of skew is difficult due to the growth of within-die process variations,[21,22] and the extent of time borrowing via PWA is limited due to the small number of short pulse widths. But we can define a new problem, in which skew and pulse width are assigned to each latch.

A similar idea emerges if we combine PWA and retiming.[23] Retiming often suffers from a large increase of latches.[24] However, when we combine it with PWA, retiming moves that involve a large increase of latches can be performed via time borrowing, and time borrowing that causes more hold time violations can be performed via retiming.

Figure 6 shows the result of combining PWA and retiming (which we'll call PWR), and of combining PWA and CSS (which we'll call PWCS). In the case of PWCS, we limited the skew to 10% of T_{min}. A clock period close to T_{min} was achieved using either PWCS or PWR. PWCS was not quite as effective in circuits b04 and b07, however. These circuits exhibited a T_{min} very far from T_{ini}, thus requiring considerable optimization, which cannot be achieved via limited skew and time borrowing. PWR was more effective in these circuits, because retiming could be performed as long as it could be applied. The average increase of latches from PWR was 13%, which was far smaller than the increase due to retiming alone; this caused about a 7% increase in the circuit area. The impact of the buffers to fix additional hold time violations from PWA or PWCS was marginal—only about a 2% increase in the circuit area.

Statistical considerations

Process-voltage-temperature (PVT) variations should be considered in pulser design. We performed Monte Carlo simulations in 45-nm technology. We applied V_{DD} variation with 0.11 V as 3σ and kept 1.1 V as the mean. We varied the temperature with 50°C as the 3σ and 75°C as the mean. The pulse width ranged from (130 − 12 ps) to (130 + 12 ps), when all three variation sources were applied.

Simply assuming a $\pm 3\sigma$ of pulse width does not represent either the worst or best case, because the difference between the pulse widths of the launching and the capturing latches—not the pulse width itself—determines the amount of time borrowing. We can address this in two different ways: including extra timing margin (in the clock period) to absorb any risk of timing violations due to pulse width variation, or directly considering variations during the design stage.

We tested several pulsed-latch circuits that used a single pulse width. We set the clock period to the delay of each circuit's timing-critical path. Then we varied the pulse width and measured the probability of each circuit satisfying its timing constraints (timing yield).[19] Next, we increased the clock period until the timing yield exceeded 90%. The clock period increment, representing the timing margin, was about a 0.5 FO4 delay, which can be compared to a 2-FO4-delay reduction in timing overhead when using latches instead of flip-flops. When we used different pulse widths, the extra timing margin varied from 0.25 to 0.6 FO4.

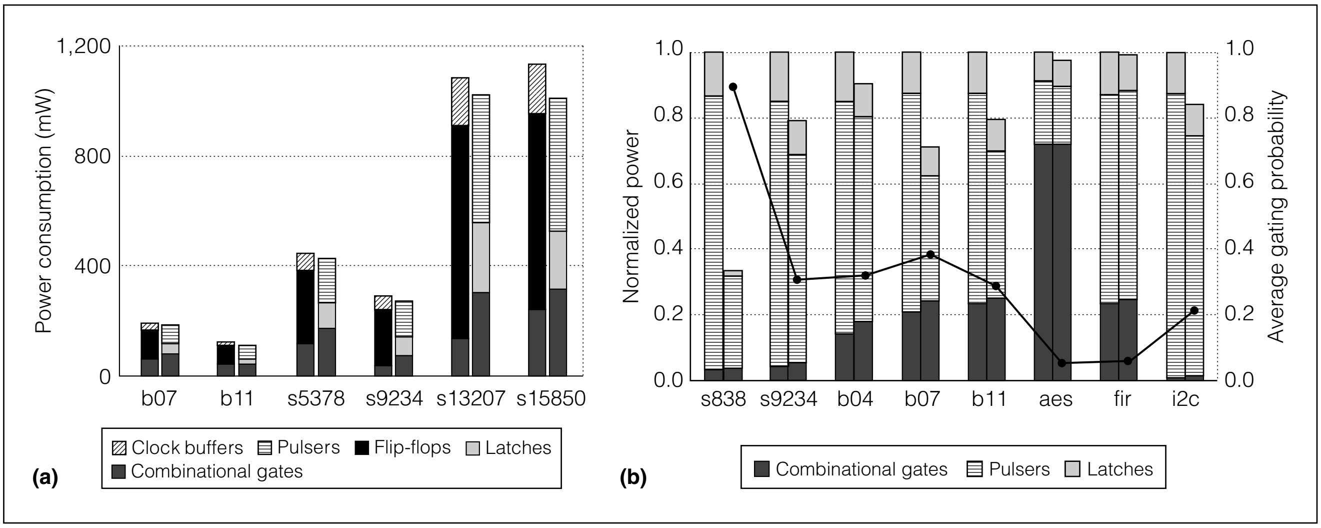

Figure 7. Power consumption of flip-flop circuits (left bars) and pulsed-latch circuits (right bars) (a), and power consumption of initial pulsed-latch circuits (left bars) and circuits employing pulser gating (right bars) (b). The bold line (plotting the dots) in (b) represents the average gating probability.

Paik, Yu, and Shin have explored PWA under pulse width variation.[19] The PWA under both circuit delay variation and pulse width variation is a difficult problem that we might pursue in the future.

Low-power considerations

A simple migration of a flip-flop circuit to a pulsed-latch version benefits power consumption,[10] even though it involves the inclusion of pulsers and delay buffers. Figure 7a illustrates the result of this migration for some test circuits in 45-nm technology, in which the power savings were 7.3% on average (minimum 4.3% and maximum 11.2%). We assume that, in pulsed-latch circuits, the pulser is a leaf-stage clock buffer as well as a pulse-generating element. Notice the difference in power consumption between flip-flops and the combined latches and pulsers. A standard D-type flip-flop consumes about 1.6 μW; a latch consumes 0.5 μW; and a pulser consumes 7.2 μW. If a single pulser drives 10 latches, the power consumption of 10 sequencing elements is reduced from 16.0 μW to 12.2 μW. Along with these savings, however, is the cost of increased power consumption in combinational gates due to the extra delay buffers.

The importance of pulsers in pulsed-latch circuits is apparent in Figure 7a. More power savings would be possible through pulser sizing, using an appropriate mix of flip-flops and pulsed latches, if a low-power pulser were developed or more latches were driven by a single pulser (although the latter might constrain placement too much).

Clock gating of pulsed-latch circuits

Clock gating is a standard practice to reduce power consumption. Either designers specify clock gating at the architectural level, or it is automatically synthesized at the RTL[25] or from a gate-level netlist.[26] A key problem in clock-gating design or synthesis is identifying a group of flip-flops that can be gated at the same time (and as often as possible).

Clock gating of pulsed-latch circuits can be implemented via pulsers (see Figure 1a), in which case it is called *pulser gating*. This implies a new problem in which we identify a group of latches that can be driven by the same pulser (and thus placed near one another) and gated at the same time.

Figure 7b shows a preliminary result of solving this pulser-gating problem.[27] The problem involves extracting a gating condition from each latch (as a Boolean expression), performing initial placement to obtain latch locations, and identifying groups of latches and inserting pulsers. Each pulser is enabled or disabled by the consensus of the gating conditions of the latches within its pulser group. Power savings are largely determined by the average gating probability. In particular, we notice (from the bold line plotting the dots in Figure 7b) that when the gating probability is high, the power in the pulsers is reduced significantly.

A PULSED LATCH IS very useful because it makes it possible to adopt a standard ASIC design flow without major changes while still enabling performance improvement and power savings. The key to pulsed latches is guaranteeing the integrity of the pulse shape; both the placement and issues regarding PVT variations and noise are important in this regard. The design of efficient pulsers and ways to adjust the pulser's strength during the design stage, such as during placement, need to be investigated. ■

References

1. D. Chinnery and K.W. Keutzer, *Closing the Gap Between ASIC & Custom: Tools and Techniques for High-Performance ASIC Design,* Springer, 2002.
2. T. Baumann, D. Schmitt-Landsiedel, and C. Pacha, "Architectural Assessment of Design Techniques to Improve Speed and Robustness in Embedded Microprocessors," *Proc. 46th Design Automation Conf.* (DAC 09), ACM Press, 2009, pp. 947-950.
3. H. Partovi et al., "Flow-through Latch and Edge-Triggered Flip-Flop Hybrid Elements," *Proc. 42nd IEEE Int'l Solid-State Circuits Conf.* (ISSCC 96), IEEE Press, 1996, pp. 138-139.
4. S. Kozu et al., "A 100 MHz 0.4W RISC Processor with 200 MHz Multiply-Adder, Using Pulse-Register Technique," *Proc. 42nd IEEE Int'l Solid-State Circuits Conf.* (ISSCC 96), IEEE Press, 1996, pp. 140-141, 432.
5. A. Scherer et al., "An Out-of-Order Three-Way Superscalar Multimedia Floating-Point Unit," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 99), IEEE Press, 1999, pp. 94-95.
6. L.T. Clark et al., "An Embedded 32-b Microprocessor Core for Low-Power and High-Performance Applications," *IEEE J. Solid-State Circuits,* vol. 36, no. 11, 2001, pp. 1599-1608.
7. N.A. Kurd et al., "A Multigigahertz Clocking Scheme for the Pentium 4 Microprocessor," *IEEE J. Solid-State Circuits,* vol. 36, no. 11, 2001, pp. 1647-1653.
8. S.D. Naffziger et al., "The Implementation of the Itanium 2 Microprocessor," *IEEE J. Solid-State Circuits,* vol. 37, no. 11, 2002, pp. 1448-1460.
9. H. Ando et al., "A 1.3-GHz Fifth-Generation SPARC64 Microprocessor," *IEEE J. Solid-State Circuits,* vol. 38, no. 11, 2003, pp. 1896-1905.
10. S. Shibatani and A.H.C. Li, "Pulse-Latch Approach Reduces Dynamic Power," *EE Times,* 17 July 2006; http://www.eetimes.com/design/power-management-design/4004576/Pulse-latch-approach-reduces-dynamic-power.
11. R. Kumar et al., "A Robust Pulsed Flip-Flop and Its Use in Enhanced Scan Design," *Proc. Int'l Conf. Computer Design* (ICCD 09), IEEE Press, 2009, pp. 97-102.
12. M. Golden et al., "A Seventh-Generation x86 Microprocessor," *IEEE J. Solid-State Circuits,* vol. 34, no. 11, 1999, pp. 1466-1477.
13. H. Lee, S. Paik, and Y. Shin, "Pulse Width Allocation and Clock Skew Scheduling: Optimizing Sequential Circuits Based on Pulsed Latches," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 29, no. 3, 2010, pp. 355-366.
14. Y.-L. Chuang et al., "Pulsed-Latch Aware Placement for Timing-Integrity Optimization," *Proc. 47th Design Automation Conf.* (DAC 10), ACM Press, 2010, pp. 280-285.
15. N.V. Shenoy, R.K. Brayton, and A.L. Sangiovanni-Vincentelli, "Minimum Padding to Satisfy Short Path Constraints," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design* (ICCAD 93), IEEE CS Press, 1993, pp. 156-161.
16. C. Lin and H. Zhou, "Clock Skew Scheduling with Delay Padding for Prescribed Skew Domains," *Proc. 12th Asia and South Pacific Design Automation Conf.* (ASP-DAC 07), IEEE CS Press, 2007, pp. 541-546.
17. Y. Sun, J. Gong, and C.-T. Chen, *Method and Apparatus for Fixing Hold Time Violations in a Circuit Design,* US patent 7278126, to Qualcomm, Patent and Trademark Office, 2007.
18. P.M. Kotecha et al., *Method of Minimizing Early-Mode Violations Causing Minimum Impact to a Chip Design,* US patent 20100042955, to IBM, Patent and Trademark Office, 2010.
19. S. Paik, L. Yu, and Y. Shin, "Statistical Time Borrowing for Pulsed-Latch Circuit Designs," *Proc. 15th Asia and South Pacific Design Automation Conf.* (ASP-DAC 10), IEEE Press 2010, pp. 675-680.
20. J.P. Fishburn, "Clock Skew Optimization," *IEEE Trans. Computers,* vol. 39, no. 7, 1990, pp. 945-951.
21. K.M. Carrig, "Chip Clocking Effect on Performance for IBM's SA-27E ASIC Technology," *IBM MicroNews,* vol. 6, no. 3, 2000, pp. 12-16.
22. S. Held et al., "Clock Scheduling and Clocktree Construction for High Performance ASICs," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design* (ICCAD 03), IEEE CS Press, 2003, pp. 232-239.
23. S. Lee, S. Paik, and Y. Shin, "Retiming and Time Borrowing: Optimizing High-Performance Pulsed-Latch-Based-Circuits," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design* (ICCAD 09), ACM Press, 2009, pp. 375-380.
24. S.S. Sapatnekar and R.B. Deokar, "Utilizing the Retiming-Skew Equivalence in a Practical Algorithm for Retiming Large Circuits," *IEEE Trans. Computer-Aided Design*

of Integrated Circuits and Systems, vol. 15, no. 10, 1996, pp. 1237-1248.

25. L. Benini and G. De Micheli, "Automatic Synthesis of Low-Power Gated-Clock Finite-State Machines," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 15, no. 6, 1996, pp. 630-643.
26. E. Arbel, C. Eisner, and O. Rokhlenko, "Resurrecting Infeasible Clock-Gating Functions," *Proc. 46th Design Automation Conf.* (DAC 09), ACM Press, 2009, pp. 160-165.
27. S. Kim et al., "Pulser Gating: A Clock Gating of Pulsed-Latch Circuits," *Proc. 16th Asia and South Pacific Design Automation Conf.* (ASP-DAC 11), IEEE Press, 2011, pp. 190-195.

Youngsoo Shin is a professor in the Department of Electrical Engineering at KAIST (Korea Advanced Institute of Science and Technology), Daejeon, Korea. His research focuses on CAD of VLSI circuits. He has a PhD in electronics engineering from Seoul National University. He is a senior member of IEEE.

Seungwhun Paik is an engineer at Synopsys in Mountain View, California. He completed the work described in this article while he was pursuing his doctorate at KAIST. His research interests include CAD for pulsed-latch ASIC design, low-power design, and structured ASICs. He has a PhD in electrical engineering from KAIST.

■ Direct questions and comments about this article to Youngsoo Shin, Dept. of Electrical Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Korea; youngsoo@ee.kaist.ac.kr.